Search help library...

Russian

Contents

1. Introduction
2. Russian example using pdfLaTeX
3. Background topics: characters and encodings
    3.1. Text files: integers and characters
    3.2. Input encoding: inputenc, UTF-8 and a change to LaTeX in 2018
    3.3. Output encoding: the fontenc package (for use with pdfLaTeX)
4. Language-specific packages and commands
5. Hyphenation
6. A final example
7. Further reading

Introduction

This article explores typesetting Russian text: enabling correct use of Cyrillic characters, such as ц, ч, ш, щ, ъ etc., and providing support for language-specific features such as hyphenation. If you are looking for instructions on how to use more than one language in a single document, for instance English and Russian, see the International language support article.

Russian example using pdfLaTeX

We'll start with the following **pdfLaTeX** example which you can open in Overleaf using the link below the code. Because Russian uses the Cyrillic script pdfLaTeX needs to ensure it not only handles the input (encoding) of Cyrillic text but also ensure the correct (output) encoding is applied to any fonts used to typeset the document. These encoding issues are discussed below.

```
\documentclass{article}
\usepackage[T2A]{fontenc}

%Hyphenation rules
%--------------------------------------
\usepackage{hyphenat}
\hyphenation{ма-те-ма-ти-ка вос-ста-нав-ли-вать}
%--------------------------------------
\usepackage[english, russian]{babel}
\begin{document}

\tableofcontents

\begin{abstract}
  Это вводный абзац в начале документа.
\end{abstract}

\section{Предисловие}
 Этот текст будет на русском языке. Это
демонстрация того, что символы кириллицы
 в сгенерированном документе (Compile to PDF)
отображаются правильно. Для этого Вы должны
установить нужный  язык (russian) и необходимую
кодировку шрифта (T2A).

\vskip12pt

\textbf{Этот текст будет на русском языке. Это
демонстрация того, что символы кириллицы в
сгенерированном документе (Compile to PDF)
отображаются правильно.}
```

```
\vskip12pt

\textit{Этот текст будет на русском языке. Это
демонстрация того, что символы кириллицы в
сгенерированном документе (Compile to PDF)
отображаются правильно.}

\section{Математические формулы}
Кириллические символы также могут быть использованы
в математическом режиме.

\begin{equation}
  S_\textup{ис} = S_{123}
\end{equation}
\end{document}
```

Open this pdfLaTeX example in Overleaf

This example produces the following output:

Содержание

1 Предисловие **1**

2 Математические формулы **1**

Аннотация

Это вводный абзац в начале документа.

1 Предисловие

Этот текст будет на русском языке. Это демонстрация того, что символы кириллицы в сгенерированном документе (Compile to PDF) отображаются правильно. Для этого Вы должны установить нужный язык (russian) и необходимую кодировку шрифта (T2A).

Этот текст будет на русском языке. Это демонстрация того, что символы кириллицы в сгенерированном документе (Compile to PDF) отображаются правильно.

Этот текст будет на русском языке. Это демонстрация того, что символы кириллицы в сгенерированном документе (Compile to PDF) отображаются правильно.

2 Математические формулы

Кириллические символы также могут быть использованы в математическом режиме.

$$S_{\text{ис}} = S_{123} \tag{1}$$

Background topics: characters and encodings

The following sections provide background material on topics related to typesetting different languages using LaTeX (mostly related to pdfLaTeX).

Text files: integers and characters

Any text file, such as a LaTeX input `.tex` file, is nothing more than a stream of numeric (integer) values which are being used as a mechanism to *represent* characters of text; consequently, processing a text file involves scanning (reading/processing) a series of integer values. However, an important question arises: *which set of characters* is actually represented by the integer values contained in a particular text file? In other words, how have those integer values been *encoded*: what is the correct (intended) "mapping" (encoding) from integers in the text file to the corresponding characters they are supposed to *represent*?

Text files can be generated within innumerable computing environments: across different countries/continents, using a multitude of different devices, operating systems and editing tools. Originators of text files could, potentially, use or apply different text encodings according to local requirements, such as language, when generating and storing the sequence of integer values chosen to represent the individual characters contained in a text file. This may work well if the generated text files stayed within compatible technical ecosystems, which use the same encoding, but what would happen when those files are transferred to completely different environments—because many text files do not contain any information which indicates the encoding used to generate them.

Clearly, the producer (originator) and consumer (user) of textual data must, somehow, agree on the encoding (mapping) being used, otherwise *encoding errors* are likely to arise due to mismatches between the integer data in the file and the set of characters it is assumed to represent. In

addition to correctly mapping text file integer values to characters, any subsequent visual display of those characters requires some form of font that is capable of providing the data (shapes, or even bitmaps) to output a visual representation of the desired characters.

Input encoding: `inputenc`, UTF-8 and a change to LaTeX in 2018

Historically, a variety of 8-bit encodings were used to generate/process text files, including LaTeX inputs. To cut short a very long story, the developers of LaTeX created the `inputenc` package to address encoding issues—allowing text files, created using various encodings, to be transferred between disparate LaTeX installations.

However, over time, users/software developers moved away from multiple 8-bit encodings to using Unicode and its UTF-8 encoding scheme, which became the de facto option for encoding text files. Prior to 2018, to process UTF-8 encoded files LaTeX document preambles included the line

```
\usepackage[utf8]{inputenc}
```

Readers might observe that the example above does not include the line `\usepackage[utf8]{inputenc}` in the document preamble: why is that? This is due to an important change to LaTeX introduced in 2018: a switch to UTF-8 as the default *input* encoding. Documents typeset with pdfLaTeX, and using UTF-8 encoded text, including those created and typeset on Overleaf, no longer *need* to include `\usepackage[utf8]{inputenc}` but is does no harm to do so. For further information see the April 2018 issue of LaTeX News and the Overleaf blog post *TeX Live upgrade—September 2019*. All text files created on Overleaf are encoded using UTF-8.

Output encoding: the `fontenc` package (for use with pdfLaTeX)

To correctly typeset characters contained within input files, those characters need to be mapped to the appropriate

output shapes (glyphs) contained in fonts used to typeset the document. This “output encoding” is handled by another package called `fontenc`.

To use `fontenc` include the following line in your document preamble, using an encoding, such as the T2A encoding, which supports the Cyrillic script:

```
\usepackage[T2A]{fontenc}
```

The following chart lists the `T2A` font encoding for Cyrillic text. It is reproduced from page 23 of the document *LaTeX font encodings* which is available on CTAN.

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	`	´	ˆ	˜	¨	˝	°	ˇ	″0x
´01x	˘	¯	˙	¸	˛	I	⟨	⟩	
´02x	“	”	‚	„	…	–	—	1	″1x
´03x	o	ı	j	ff	fi	fl	ffi	ffl	
´04x	␣	!	"	#	$	%	&	’	″2x
´05x	(	)	*	+	,	-	.	/	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	:	;	<	=	>	?	
´10x	@	A	B	C	D	E	F	G	″4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	″5x
´13x	X	Y	Z	[	\	]	^	_	
´14x	‘	a	b	c	d	e	f	g	″6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	″7x
´17x	x	y	z	{	|	}	~	-	
´20x	Ґ	Ғ	Ҕ	Ђ	Ћ	Ҋ	Ӂ	Ӆ	″8x
´21x	Ӣ	Ӄ	Ҝ	Ќ	Ӕ	Ӊ	Ҥ	S	
´22x	Ө	Ҫ	Ӱ	Ү	Ұ	Ҳ	Ӌ	Ҷ	″9x
´23x	Ӌ	Є	Ә	Њ	Ё	№	¤	§	
´24x	ґ	ғ	ҕ	ђ	ћ	ҋ	ӂ	ӆ	″Ax
´25x	ӣ	ӄ	ҝ	ќ	æ	ӊ	ҥ	s	
´26x	ө	ҫ	ӱ	ү	ұ	ҳ	ӌ	ҷ	″Bx
´27x	ӌ	є	ə	њ	ё	„	«	»	
´30x	А	Б	В	Г	Д	Е	Ж	З	″Cx
´31x	И	Й	К	Л	М	Н	О	П	
´32x	Р	С	Т	У	Ф	Х	Ц	Ч	″Dx
´33x	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	
´34x	а	б	в	г	д	е	ж	з	″Ex
´35x	и	й	к	л	м	н	о	п	
´36x	р	с	т	у	ф	х	ц	ч	″Fx
´37x	ш	щ	ъ	ы	ь	э	ю	я	
	″8	″9	″A	″B	″C	″D	″E	″F	

Language-specific packages and commands

To extended the default LaTeX capabilities, for proper hyphenation and translating the names of the document elements, import the `babel` package and use the `russian` language option:

```
\usepackage[russian]{babel}
```

As you may see in the example in the introduction, instead of "Abstract" and "Contents" the Russian words "Аннотация" and "Содержание" are used. Cyrillic characters can also be used in mathematical mode.

If you need to include more than one language, for instance, Russian and English, see the International language support article to learn how to achieve this. A great deal of useful background information can be found in the document Russian language module for Babel.

Hyphenation

To achieve linebreaks, some words have to be hyphenated: broken up into syllables separated by a `-` (hyphen), allowing the word to continue on the next line. For example, математика could become мате-мати-ка. The `babel` package usually does a good job of hyphenation but, occasionally, you may need to define hyphenation points manually, which you can do using these commands in the preamble:

```
\usepackage{hyphenat}
 \hyphenation{мате-мати-ка восста-навливать}
```

`\usepackage{hyphenat}` imports the `hyphenat` package and the second line is a list of space-separated words with defined hyphenation points. If you want to prevent automatic hyphenation of a particular *`word`*, use the command `{\nobreak` *`word`*`}` within your document.

A final example

This final example configures LaTeX so that the Russian text can be typeset using pdfLaTeX, XeLaTeX and LuaLuaTeX—to

switch compilers on Overleaf, see the article Changing compiler.

This example uses the `\iftutex` command from the `iftex` package to detect whether a Unicode-aware engine (LuaTeX or XeTeX) is being used to compile the LaTeX code (document). If so, Google's OpenType Noto fonts are configured via the `fontspec` package and used to typeset the document. If a non-Unicode-aware engine, such as pdfTeX, is detected then the LaTeX code (document) is typeset using old 8-bit font technologies (based on Adobe Type 1).

The following code opens on Overleaf and defaults to compiling with XeLaTeX. To use a different compiler, see the article Changing compiler.

```
\documentclass{article}
\usepackage{iftex}
\iftutex
% For LuaTeX or XeTeX Use Google's
% OpenType Noto fonts for typesetting
% Russian text
\usepackage{fontspec}
\defaultfontfeatures{Ligatures={TeX}}
\setmainfont{Noto Serif}
\setsansfont{Noto Sans}
\setmonofont{Noto Sans Mono}
\else
% For pdfTeX we must use old
% 8-bit font technologies
\usepackage[T2A]{fontenc}
\fi
%Hyphenation rules
\usepackage{hyphenat}
\hyphenation{ма-те-ма-ти-ка вос-ста-нав-ли-вать}
\usepackage[english, russian]{babel}
\begin{document}

Этот текст будет на русском языке. Это демонстрация
того, что символы кириллицы в сгенерированном
документе отображаются правильно.
```

```
\vskip12pt

\textbf{Этот текст будет на русском языке. Это
демонстрация того, что символы кириллицы в
сгенерированном документе отображаются правильно.}

\vskip12pt

\textit{Этот текст будет на русском языке. Это
демонстрация того, что символы кириллицы в
сгенерированном документе отображаются правильно.}

\vskip12pt

\texttt{Этот текст будет на русском языке.}
\end{document}
```

Open this example on Overleaf

Using XeLaTeX, this example produces the following output:

Этот текст будет на русском языке. Это демонстрация того, что символы кириллицы в сгенерированном документе отображаются правильно.

Этот текст будет на русском языке. Это демонстрация того, что символы кириллицы в сгенерированном документе отображаются правильно.

Этот текст будет на русском языке. Это демонстрация того, что символы кириллицы в сгенерированном документе отображаются правильно.

```
Этот текст будет на русском языке.
```

Further reading

For more information see

- Supporting modern fonts with XeLaTeX
- Typesetting quotations and quotation marks
- International language support
- Chinese
- French
- German

- Greek
- Italian
- Japanese
- Korean
- Portuguese
- Arabic
- Spanish
- Russian language module for Babel
- History of the T2A, T2B and T2C Cyrillic encodings
- The not so short introduction to LaTeX 2ε
- LaTeX/Internationalization on WikiBooks
- LaTeX/Special_Characters on WikiBooks

Documentation Home

Overleaf Learning Center

Overleaf guides

Overleaf user documentation

Overleaf group administration

Overleaf Commons administration

Overleaf on-premises

LaTeX basics

Creating your first LaTeX document

Paragraphs and new lines

Bold, italics and underlining

Lists

Errors

Mathematics

Mathematical expressions

Subscripts and superscripts

Brackets and Parentheses

Matrices

Fractions and Binomials

Aligning equations

Operators

Spacing in math mode

Integrals, sums and limits

Display style in math mode

List of Greek letters and math symbols

Mathematical fonts

Using the Symbol Palette in Overleaf

Figures and tables

Tables

Positioning Images and Tables

Lists of Tables and Figures

Drawing Diagrams Directly in LaTeX

TikZ package

Document structure

Sections and chapters

Table of contents

Cross referencing sections, equations and floats

Indices

Glossaries

Nomenclatures

Management in a large project

Multi-file LaTeX projects

Hyperlinks

Formatting

Lengths in LaTeX

Headers and footers

Page numbering

Paragraph formatting

Line breaks and blank spaces

Text alignment

Page size and margins

Single sided and double sided documents

Multiple columns

Counters

Code listing

Code Highlighting with minted

Using colours in LaTeX

Footnotes

Margin notes

References and citations

Bibliography management with bibtex

Bibliography management with natbib

Bibliography management with biblatex

Bibtex bibliography styles

Natbib bibliography styles

Natbib citation styles

Biblatex bibliography styles

Biblatex citation styles

Fonts

Font sizes, families, and styles

Font typefaces

Supporting modern fonts with XeLaTeX

Presentations

Beamer

Powerdot

Posters

Commands

Commands

Environments

Languages

Multilingual typesetting on Overleaf using polyglossia and fontspec

Multilingual typesetting on Overleaf using babel and fontspec

International language support

Quotations and quotation marks

Arabic

Chinese

French

German

Greek

Italian

Japanese

Korean

Portuguese

Russian

Spanish

Field specific

Theorems and proofs

Chemistry formulae

Feynman diagrams

Molecular orbital diagrams

Chess notation

Knitting patterns

CircuiTikz package

Pgfplots package

Typesetting exams in LaTeX

Attribute Value Matrices

Class files

Understanding packages and class files

List of packages and class files

Writing your own package

Writing your own class

Advanced TeX/LaTeX

In-depth technical articles on TeX/LaTeX